組込みソフトウェアバグ管理手法の紹介
組込みソフトウェア開発における品質向上の勧め
［バグ管理手法編］

2013年 6月13日

独立行政法人 情報処理推進機構(IPA)
技術本部 ソフトウェア高信頼化センター(SEC)
IPA/SEC 調査役 三原 幸博

バグ管理手法解説

2013年 3月 8日発行
書籍：
A5判、80ページ
定価500円（税込）

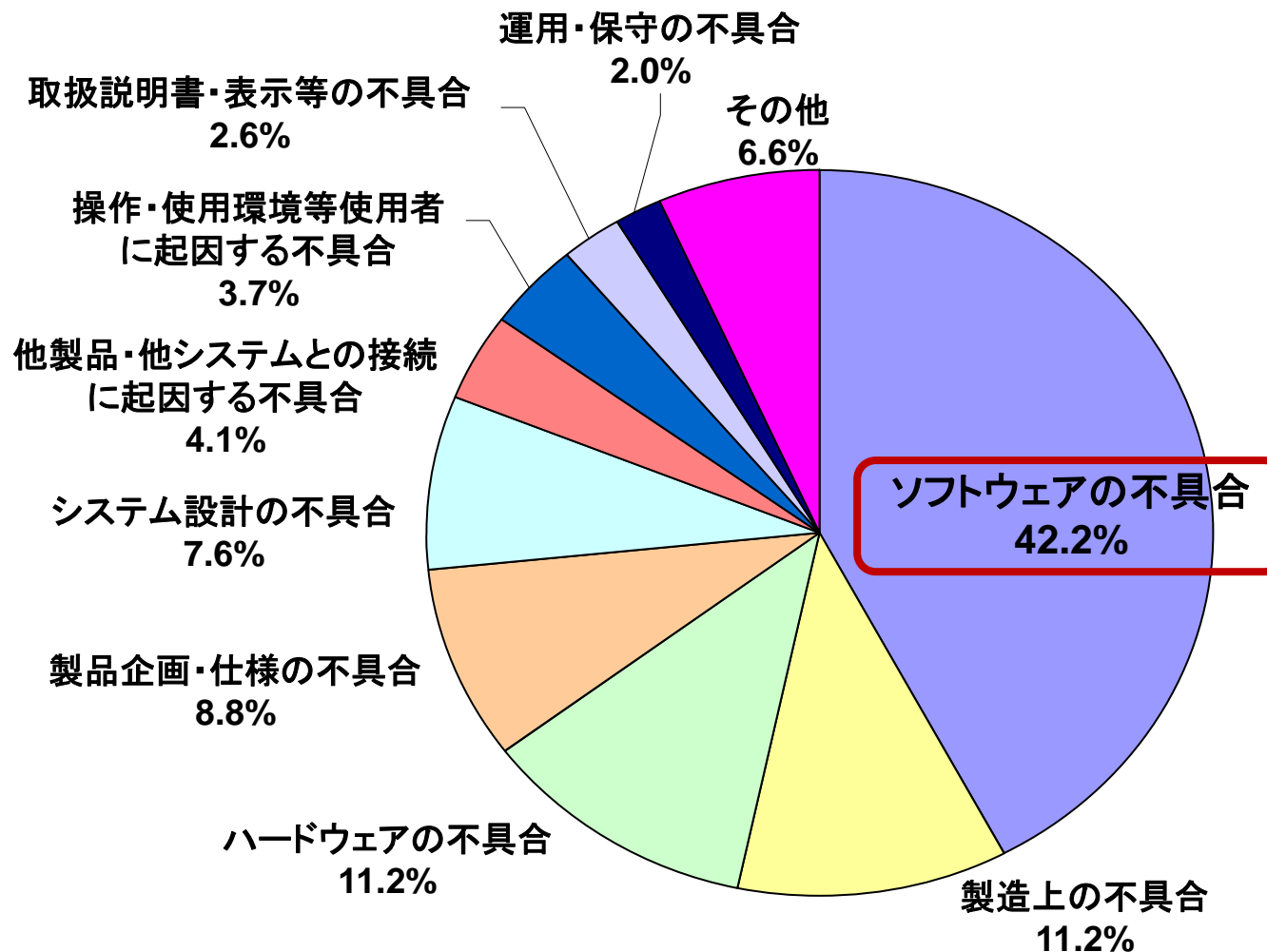
pdf版：
無償提供

1. 背景

- 組込みソフトウェア開発におけるバグ数が増加
  - ✓ ソースコード規模や連携する他ソフトウェアの増加で、テスト工程で発見されるバグ数が増加
- 品質確保のコストを上げずに、市場出荷後のバグを削減することが社会的要請
- バグ管理の重要性の拡大
  - ✓ 対策漏れの防止、潜在バグの削減と市場流出防止、対策の効率化とスピードアップ等のためのバグ管理が必要
  - ✓ 多人数での組織開発がメインとなっており、バグの発見から修正の割り振り、原因究明、修正、確認、承認など一連のバグ管理プロセスが必要
  - ✓ さらに、バグ管理を組織としての課題の抽出と改善のためのPDCAに繋ぐことによる開発能力向上が重要

組込みシステムの不具合発生状況

出荷後の不具合原因（製品ベース）

2011年版組込みソフトウェア産業実態把握調査報告書

製品ベースでも、不具合原因で最も多いのがソフトウェア不具合

問題意識

バグ管理の重要性にもかかわらず・・・・・

- 標準的な管理項目、目的、プロセスなどを説明した日本語の標準的なガイドがない
  - ✓ 新規にバグ管理を実施する場合に、管理項目の検討に時間がかかる
  - ✓ 個々のバグ管理項目の目的の理解が不十分で、入力データがばらつく（正しく入力されない）
- 何をバグとするのか、バグ1件はどのようにカウントするのかなど、基本的なバグの測定方法の指針もない

  例えば
  - ✓ コーディング規約違反はバグなのか
  - ✓ 現象は複数でも原因が1箇所の場合、1件とカウントするか
- バグに関連する用語も様々で混乱もみられる

海外の参考文献

- 海外では、CMU／SEIやIEEEに次のような資料が存在
  - [CMU/SEI-92-TR-022]
    Technical Report CMU/SEI-92-TR-022 Software Quality Measurement: - A Framework for Counting Problems and Defects
  - [IEEE 1044]
    IEEE Std. 1044.1-1995　IEEE Guide to Classification for Software Anomalies
    IEEE std. 1044-2009　IEEE Standard Classification for Software Anomalies

※これらに相当する日本語の資料は存在しない

※日本でも企業の中では整理されているが、各企業の中に閉じている

2. 本ガイドの目的

目的

日本の組込みシステム開発における標準的なバグ管理方法を示し、バグ管理の底上げ、バグデータ測定の平準化を通して組込みソフトウェア（製品）の品質向上を図る

具体的には

バグ管理環境を整備する品質管理推進者、プロジェクトマネージャ、及びバグ情報を入力するテスト実施者、開発者がバグ管理の重要性を認識してバグ管理のプロセスを実行できるようにする

（参考）ESxRとの関係

SEC
Software Reliability
Enhancement Center

ESQR（品質作り込みガイド）
ESPR（開発プロセスガイド）
What
How
【プロセス定義】【品質指標】
本ガイドの
対象範囲
経営者
開発担当
要求定義
要求仕様書の評価指標
アーキテクチャ設計
設計書の評価指標
ESDR（設計ガイド［事例編］）
ソフトウェア詳細設計
コーディング
システムの品質評価指標
システムテスト
テスト作業の評価指標
結合・統合テスト
ESTR（テスト事例集）
単体テスト
ソースコードの品質評価
ESCR（コーディング作法ガイド）
マネージャ
マネジメント指針
ESMR（プロジェクトマネジメントガイド）

ESxR との関係(続き)

SEC
Software Reliability
Enhancement Center

開発計画・管理
ソフトウェア
設計・開発
レビュー
&テスト
ESPR
ESMR
ESQR
ESDR
ESCR
テスト編
プロセス改善
品質制御
設計・実装手法改善
方針の連携
バグ管理
手法
バグ管理
情報
記録・参照
傾向分析
修正作業
原因分析

3. バグ管理の目的

目的

- バグの修正
- 製品リリース時の残存バグの有無の把握
- バグの検出状況によるソフトウェアの品質
- バグの分析によるソフトウェア開発のカイゼン

効果

- 管理項目をデータベース化し統一して扱うことで、バグの追跡が自動化され、正確な分析が可能になる
- 開発者は、プロジェクトや顧客の夫々にとって重要な観点から修正を進めることが可能になる
- 発見から解決まで、全てのライフサイクルを通したバグ管理ができ、対応の抜け漏れの防止が可能になる

4. バグに関連する用語について

JIS Z 8115の解説

前略～

バグには以下のような**3種類の意味**が込められているとする意見もある。すなわち“人間の思考過程における人の誤りに代表される**不完全性又は過失**”、または“そのような過失の結果として設計に表現されてしまった記述上の不十分性、不統一性、不完全性、矛盾性のような**欠陥**”と“そのような記述上の欠陥が原因となって、ソフトウェアが稼動している時に外部から観測される機能、性能、使い易さ、保守のし易さなどの**不具合**”である。

～後略

バグの定義：

ソフトウェアで設計者の認識有無にかかわらず、すべての成果物において要件定義の誤り、仕様設計の誤り、プログラミングの誤り、システム構築の誤りなどにより「期待される結果」と乖離があるために、何かしらの対策・対応が必要と考えられる**現象またはその原因**。

特に現象と原因を区別する場合は、現象を示す時は「バグ現象」と呼び、原因を示す時は「バグ原因」と呼ぶ。

＊IEEE982.1-2005の“defect”（欠陥）に対応している

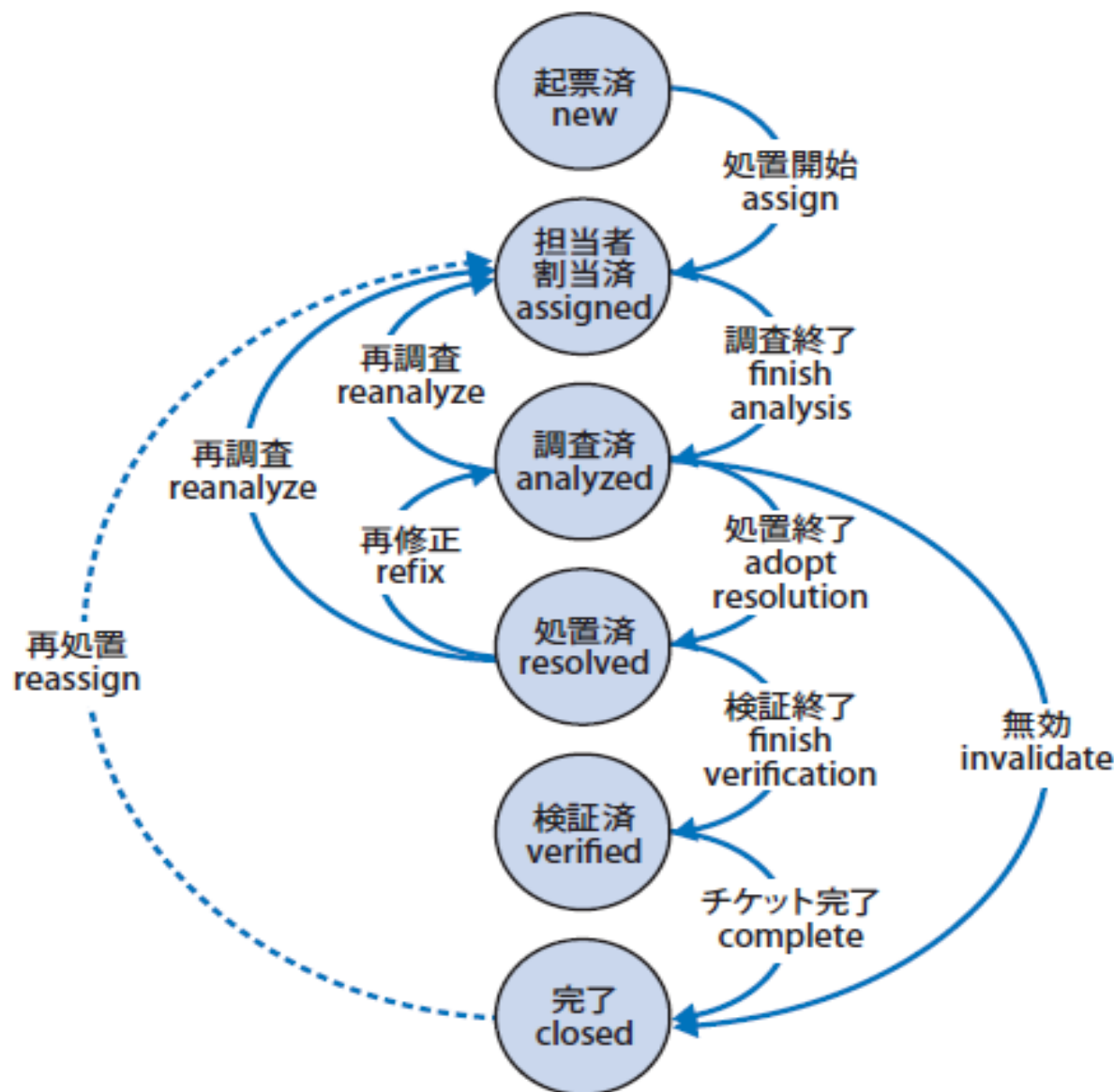
5. バグ管理プロセス

バグ管理プロセス:

バグが発見されてから、分析や処置が行われ、対応が完了したことが確認されるまでの一連の活動のこと

※ バグに着目した表現として、発見から問題解決まで、バグの処置状態の変遷を、バグのライフサイクルともいう

⇒ バグ管理プロセスは、バグ管理の基本であり、バグ管理を始めるまでに定義する

バグ管理システムにおける管理状態の遷移

6. バグ管理内容と管理項目

バグの管理のための標準的な項目の一覧を提示

特徴は次の通り

- 項目選定では、CMU/SEI-92-TR-022やIEEE1044などのグローバル標準との整合性を考慮
- 項目の説明では、バグ情報入力者が目的を理解できるように、各項目ごとに目的の説明を提示
- 項目名は英語も併記(グローバルな開発体制を考慮)

利用方法

- 組織として必須管理項目を定める
- プロジェクトに合せて管理項目を選択する

バグ管理項目 -1

項目名		説明・属性	本項目の目的
日本語	英語		
管理番号	ID	管理のための番号。	バグをユニークに識別、管理するための情報。
概要 (タイトル)	title	概要を1行程度で示す。	・バグの内容を大まかにつかむための情報。 ・タイトルの付け方をルール化すると検索しやすくなる。
プロジェクト名	project name	対象プロジェクトの名前を記述する。	・プロジェクトごとにバグを1種として取りまとめるための情報。 ・バグ情報をプロジェクト単位で管理する場合は無くてもよい。
重要度 ※製品／顧客視点	severity	重要度を、製品または顧客の視点で示した分類。 (例)S：最重要、A：重要、B：、C：重障害、中障害、軽障害	・バグが与える影響の度合いを分類で示すことにより、分類別にバグを管理して、絞り込むための情報。 ・品質状況把握、修正の優先順位付け、出荷判定に利用するなどの顧客視点での判断を行うために用いる情報。

バグ管理項目（担当者、日付）-1

項目名		説明・属性	本項目の目的
日本語	英語		
発行者	opened by	本票の作成者。	バグ票（本票）を登録記載した人を明確にするための情報。
発行日	date opened	本票の発行日。 必要であれば、時間まで記録する。	・本票の発行日を明確にするための情報。 ・バグ発行日から完了日までの日数から対応期間や滞留期間などを算出し、バグ対応時間実績を把握する。これにより、対応状況や今後の見通し（判断）、次プロジェクトの計画立案への情報などとして活用することが可能となる。
発見者	observed by	バグの発見者。 発行者と同じ場合、同じ名前を記入するか、省略するかなど、プロジェクトにより運用が異なる。	・バグの発見者を明確にするための情報。 ・バグ原因調査（再現）の際、内容欄に書かれた情報だけでは不明、あるいは再現できない場合、プロジェクトリーダなどが問い合わせを行うための情報として用いることもある。

バグ管理項目（調査結果）－1

項目名		説明・属性	本項目の目的
日本語	英語		
発生原因	cause	バグ発生の原因分析結果。（調査担当者や、処置担当者が分析した発生原因）	・バグ発生の原因を記録するための情報。横展開にも利用される。 ・調査担当者と処置担当者とが異なることが多いため、調査担当と処置担当の原因分析結果を別々に記録したい場合、別項目として定義するとよい。個別に記録が必要でない場合は、１項目として定義、記録するとよい。
原因箇所	defect found in	・バグが発生した原因を含むソフトウェア成果物。 ・仕様書であれば仕様書名（ファイル名）とその頁、行数など。ソースコードであれば、ファイル名、関数名、行数など。	・バグ発生の原因箇所を明示するための情報。 ・バグ分析にも用いる。

7. バグカウントの指針

- ➢バグの数は、様々な指標に利用されるが、バグ１件のカウント方法が定まっていないと、指標値が信用できなくなる
- ➢バグ１件のカウント方法は、開発組織やプロジェクトチーム内で揃えることが重要
- ➢本ガイドで推奨する数え方の原則は次の通り

● **バグ１件のカウント方法の原則**

「設計仕様書やソースコード等のソフトウェア成果物に含まれるバグの原因部分について１件とする」

（参考資料：SQiPシンポジウム2010 併設チュートリアル ソフトウエア品質データ分析の作法　野中誠）

8. バグ分析

バグの分析は、主に次の目的で行なう。

- **現ソフトウェア品質の推定、後工程・最終品質の予測**
  - ✓ バグの分析結果から、後工程や稼働時の最終品質を予測する。
  - ✓ それにより後工程の作業内容や作業スケジュールの調整を行い、最終品質が要求レベルに達するようにする。
- **開発プロセスのカイゼン**
  - ✓ 振り返りによる問題の特定と対策を実施する。
  - ✓ バグの作り込み要因や前工程で発見出来なかった要因を分析し、ソフトウェア開発プロセスの問題を特定し、カイゼンにつなげる。

分析例

➢ **赤い丸印の変化点に注目して調べる**

Ver.2.0に向けての
新規機能開発

機能追加が発生。
Ver.2.0に組み込む
機能のみ実装

Ver.2.0リリース

計画　新規機能開発　結合テスト　機能追加　総合テスト

180
160
140
120
100
80
60
40
20
0

再テストを実施して
いない可能性あり

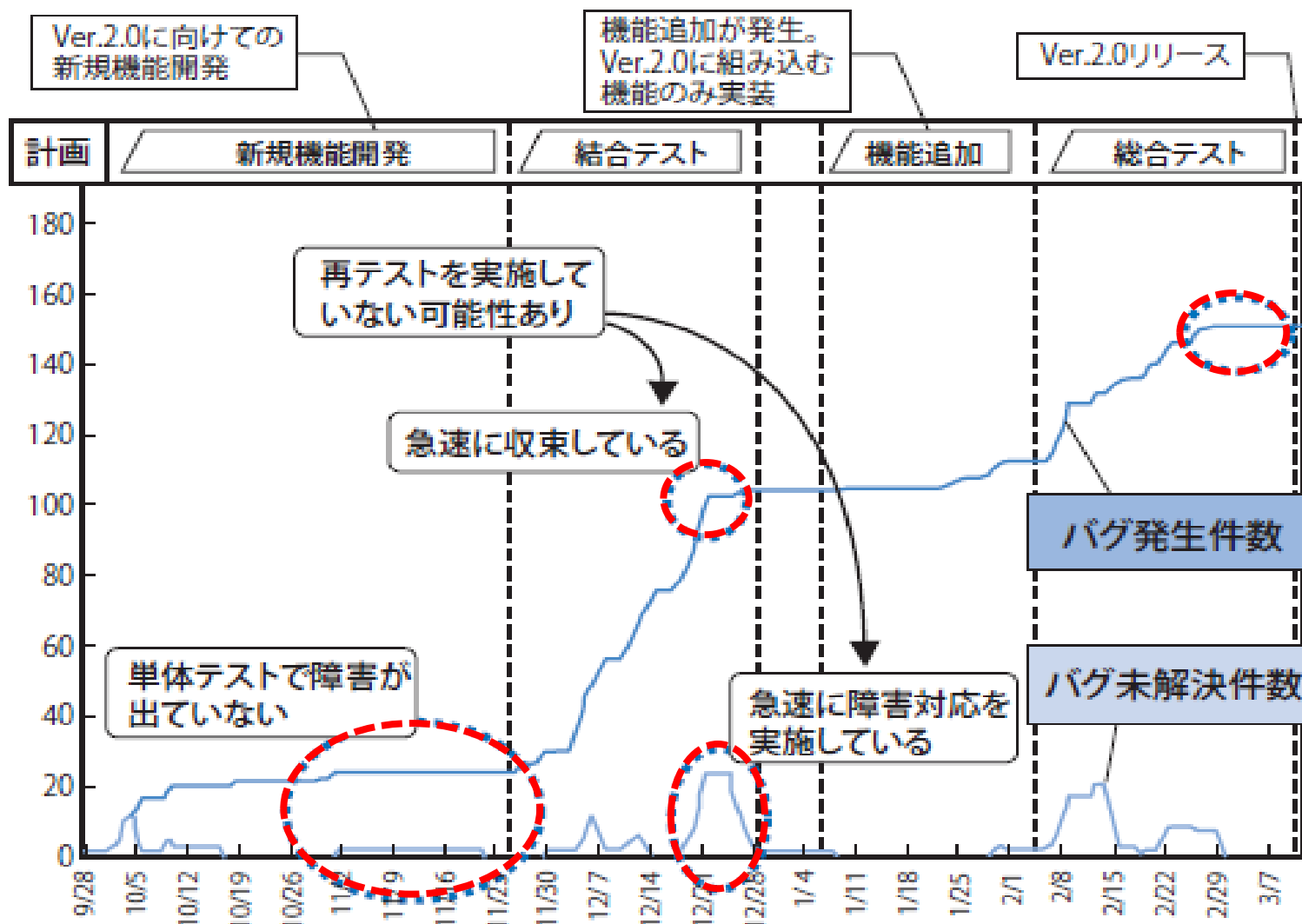
急速に収束している

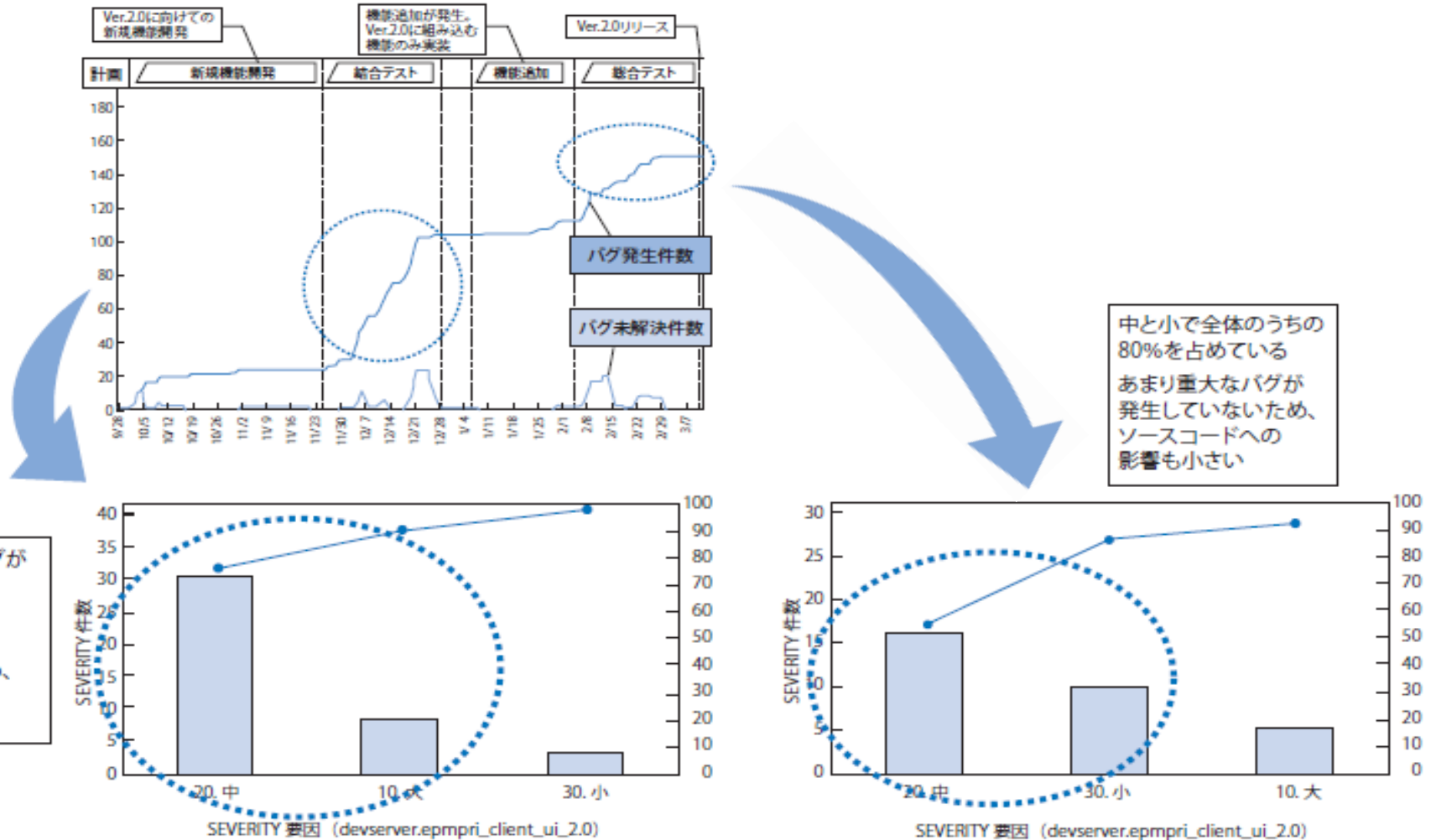
単体テストで障害が
出ていない

急速に障害対応を
実施している

バグ発生件数

バグ未解決件数

9/28 10/5 10/12 10/19 10/26 11/2 11/9 11/16 11/23 11/30 12/7 12/14 12/21 12/28 1/4 1/11 1/18 1/25 2/1 2/8 2/15 2/22 2/29 3/7

分析例（重大度）

➢ 後半のバグ数は多いが重大バグが少ないため、品質は安定してきていると考えられる。

9. 組込みシステムにおけるバグ管理の勘所

- 実例に基づく、いくつかの勘所をコラムとして提示
  - どこからバグと呼ぶか
  - 未解決バグの棚卸
  - ブロッキングバグの扱い
  - 担当振り分け方法
  - Duplicateバグ発生の抑制
  - 類似バグの一掃
  - なぜなぜ分析
  - バグ収束とバグ曲線
- また、大規模開発の場合のバグ管理の参考として、エンタープライズ系のバグ管理をコラムとして提示

IPAセミナー（ESxRシリーズ関連）

2013年6月14日（金）@会議センター 会議室A+B

第4部 11:00～12:30

- **組込みソフトウェア向け設計ガイド ESDR[事例編]紹介**
  - 株式会社アックス 浜田直樹
  - 三菱電機メカトロニクスソフトウエア(株) 岩橋正実
  - 横河医療ソリューションズ(株) 楢木野公彦

第5部 13:00～14:30

- **組込みソフトウェアテスト事例集の紹介**
  - IPA/SEC研究員 石田 茂
  - 日本電気通信システム株式会社 羽田 裕

第6部 15:00～16:30

- **組込みソフトウェア開発におけるバグ管理ガイドの紹介**
  - IPA/SEC調査役 三原 幸博

ご清聴ありがとうございました

＜PR＞

仕事につながる
国家試験。

「iパス（ITパスポート試験）」は
ITに関する基礎知識を問う国家試験です。
IT化された社会で働くすべての方に
必要な基本的能力を証明できます。

http://www.jitec.ipa.go.jp/ip/